JULY/AUGUST 1987 ISSUE # 13

HUNTER VALLEY VZ JOURNAL



PRODUCED BY HUNTER VALLEY VZ U.GROUP

BELIEVE IT OR NOT :-
My son (17) takes great delight in telling everybody that he taught his Dad all he knows about computers. I wish it was'nt half true. Anyone like to adopt him as he'll be up for adoption soon.

CONGRATULATIONS JEANETTE AND DAVE MITCHELL ON YOUR NEW ARRIVAL BABY BOY -- ROBERT THOMAS MITCHELL

```
10 '*********************************************************
20 '* HINTS AND TIPS FOR PSEUDO INPUT@, SIMPLE TIME DELAY NOT *
30 '* USING A LOOP AND USING THE ELSE STATEMENT - BY JOE LEON *
40 '*********************************************************
50 :
60 CLS:GOSUB100:GOTO10
100 CLS:PRINT@249,"Y";:PRINT@230,;:INPUT"ARE YOU SURE Y/N ";A$
120 IFA$="N"THEN900ELSEIFA$="Y"THEN800ELSE100
800 PRINT@360,"YOU PRESSED  'Y'":SOUND30,1;20,1;0,5:RETURN
900 PRINT@360,"YOU PRESSED  'N'":SOUND30,1;20,1;0,5:RETURN
```

Most persons with a computer can learn some programming techniques just by studying someone else's programs. It's how I learned and am still learning. And now to the hints.

INPUT@ :-
The VZ does not have this command, but it is easy to simulate.

```
100 PRINT@230,;:INPUT"ARE YOU SURE Y/N ";A$
```

The trick is to put a 'PRINT@230,;:' before your INPUT statement.

FLASHING CURSOR OVER Y OR N :-
If you typed in the little program and RUN it you will notice that the CURSOR is flashing over the 'Y'. Again it's simple to achieve. The tip is to PRINT the 'Y' before the INPUT command. Study Line 100 on how it's done.

TIME DELAY :-
Although the VZ has a wiped out TIME$ command, but unfortunately no processing routine for it. So instead of using a loop there is a another way. EG :- SOUND0,5 . By using a '0' for the first SOUND parameter we have a simple time delay. See Lines 800 and 900 for examples. The second SOUND parameter can be from 1 to 9, with 1 giving shortest delay, 9 the longest.

Using the ELSE Statement :-
By using ELSE in Line 120 we have compacted to one line what normally would take two or three. EG :-

```
120 IFA$="N"THEN900ELSEIFA$="Y"THEN800ELSE100
```

```
120 IF A$="N" THEN 900
130 IF A$="Y" THEN 800
140 GOTO 100
```

As you can see by using ELSE we eliminated two Lines, saved memory space and the program will RUN faster. The program as it is will not accept any INPUT exept for a 'Y' or 'N'. Lines 100 and 120 are heart of routine while the other lines demonstrate routines action.

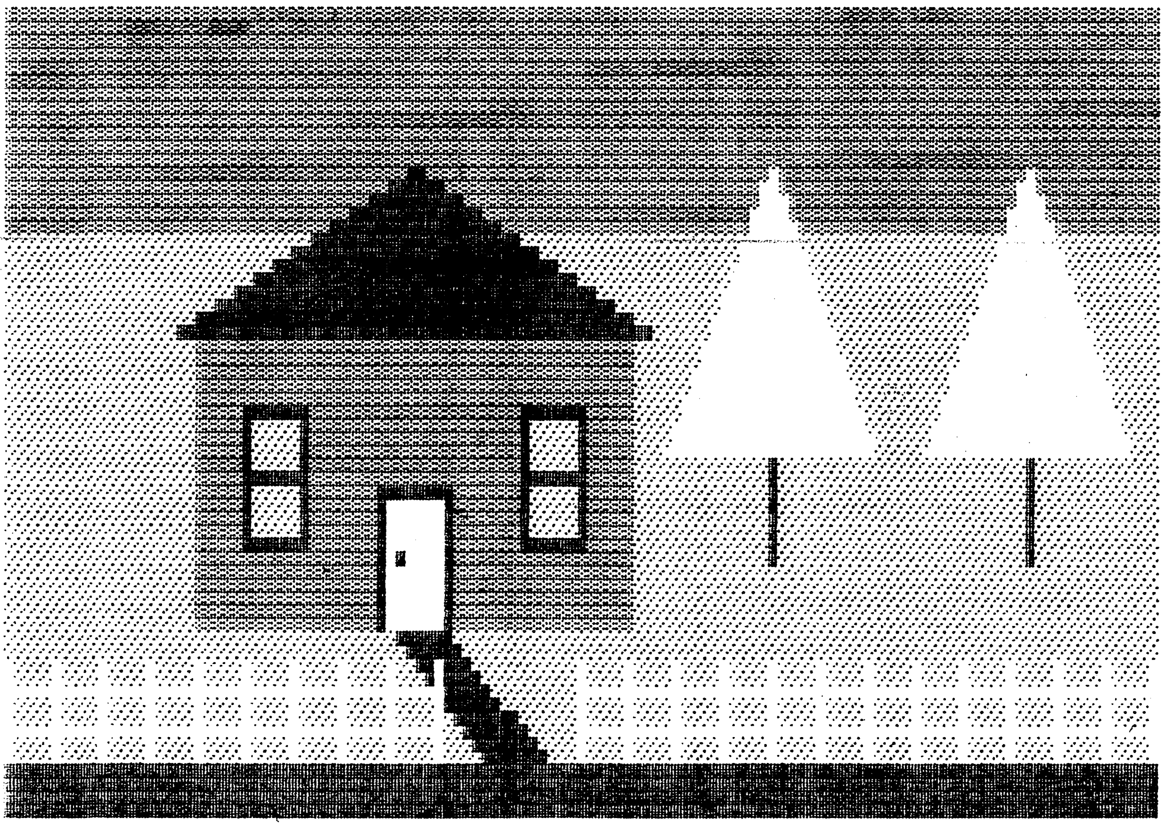
The COLON (:)
As you may have noticed the colon is by itself in line 50. It can serve two functions used that way.

1) To reserve Line/s for future use.
2) To use as a divider between routines in your programs. See the HI-RES house LISTING in this issue for a perfect example. A "'" (Short form REM) could also be used, but it takes three bytes as against one for the COLON.

PROGRAMME EXPLANATION OF HI-RES DRAWING ROUTINES

Lines 20 to 80 describe the programme and gives the choice of gable or round roof on the house.
Line 80 sets the value of 'R' at '0' for gable roof or '1' for round roof.
Line 100 sets graphic mode.
Lines 110 to 170 establishes background colour.
Lines 200 to 250 draws path.
Line 220 causes the path to slant to the left.
Lines 255 to 290 draws front wall of house.
Line 295 sets colour of roof.
Line 300 selects roof shape.
Lines 305 to 330 draws gable roof.
Lines 340 to 375 draws a round roof using the trigonometrical functions of sine (SIN) and cosine (COS). As the trig. functions in the VZ are in radians the angle value must be divided by 57.2957 in Line 345 prior to applying the function.
Lines 380 to 490 draws the fence.
Lines 500 to 600 draws the door.
Lines 610 to 820 draws the windows.
lines 860 to 950 draws th trees.
Line 1000 keeps the VZ in graphics mode by providing an endless loop to prevent the VZ returning to text mode.
Line 2000 provides an option for disk drive owners to save the Hi-Res picture of the house to disk. Just put a REM in front of GOTO in Line 1000.

NOTE :- As the program consists of quite a few subroutines you could type in line 1000 after line 180 was typed in. Then you could RUN the program to see what the first subroutine does and do the same for each subsequent subroutine. It may help you to understand how SET(X/Y) works.



```
2 '****************************************************************
4 '*  HI-RES DRAWING ROUTINES FOR GABLE OR ROUND ROOF HOUSE  *
6 '*  BY DON ISLES - MEMBER OF HUNTER VALLEY VZ USERS' GROUP  *
8 '****************************************************************
10 :
20 CLS:PRINT:PRINT"  THIS PROGRAMME IS DESIGNED TO"
30 PRINT" DRAW A HOUSE, FENCE AND TREES.
40 PRINT" YOU HAVE CHOICE OF ROOF SHAPE,  GABLE OR ROUND.
50 PRINT:PRINT
60 PRINT" PRESS  RETURN  FOR GABLE ROOF"
70 PRINT:PRINT"                    OR":PRINT
80 INPUT" PRESS 1 THEN RETURN FOR ROUND  R00F  ";R
90 :
95 '" DRAW BACKGROUND
100 MODE(1)
110 COLOR 3
120 FOR Y=0 TO 60
130 IF Y=17 THEN COLOR 2
140 IF Y=57 THEN COLOR 4
150 FOR X=0 TO 120
160 SET(X,Y)
170 NEXT X:NEXT Y
180 :
195 REM'" DRAW  PATH
200 B=40:C=5
210 FORY=47TO56
220 B=B+1:C=C+.15
230 FORX=BTOB+C
240 SET(X,Y)
250 NEXTX:NEXTY
251 :
252 REM'" DRAW  WALL
255 COLOR3
260 FORY=46TO25STEP-1
270 FORX=20TO65
280 SET(X,Y)
290 NEXTX:NEXTY
291 :
292 REM'" DRAW  ROOF
295 COLOR4
300 IF R=1 THEN 340
305 B=16:C=69
310 FOR Y=24 TO 12 STEP -1
315 B=B+2:C=C-2
320 FOR X=B TO C
325 SET(X,Y)
330 NEXT X:NEXT Y
335 GOTO 379
340 FOR M=20 TO 90 STEP 2
345 N=M/57.2957
350 L=COS(N)*27
355 Y=SIN(N)*14
360 FOR X=0 TO L
365 SET(42+X,29-Y)
370 SET(43-X,29-Y)
375 NEXT X:NEXT M
377 :
379 REM'" DRAW  FENCE
380 COLOR1:FORY=51 TO 54 STEP 3
385 FOR X=0 TO 44
390 SET(X,Y)
395 NEXT X:NEXT Y
400 FOR Y=51 TO 54 STEP 3
410 FOR X=61 TO 120
420 SET(X,Y)
430 NEXT X:NEXT Y
440 FOR X=0 TO 45 STEP 5
450 FOR Y=49 TO 56
460 SET(X,Y)
470 NEXT Y:NEXT X
475 FOR X=60 TO 120 STEP 5
480 FOR Y=49 TO 56
485 SET(X,Y)
490 NEXT Y:NEXT X
495 :
499 REM'" DRAW  DOOR
500 FOR Y=46 TO 37 STEP -1
510 FOR X=40 TO 45
520 SET(X,Y)
530 NEXT X:NEXT Y
535 COLOR 4
540 FOR X=39 TO 46 STEP 7
550 FOR Y=46 TO 37 STEP -1
560 SET(X,Y)
570 NEXT Y:NEXT X
580 FOR X=39 TO 46
590 SET(X,36)
595 NEXT X
600 SET(41,41)
602 :
605 '" DRAW WINDOWS
610 COLOR 2
620 FOR Y=31 TO 39
630 FOR X=26 TO 30
640 SET(X,Y)
650 SET(X+29,Y)
660 NEXT X:NEXT Y
690 COLOR 4
700 FOR Y=30 TO 40 STEP 5
710 FOR X=26 TO 30
720 SET(X,Y)
730 SET(X+29,Y)
740 NEXT X:NEXT Y
780 FOR X=25 TO 31 STEP 6
790 FOR Y=30 TO 40
800 SET(X,Y)
810 SET(X+29,Y)
820 NEXTY:NEXTX
```

```
850 REM'"[illegible]
860 FOR Y=41 TO 34 STEP -1
870 SET(80,Y)
875 SET(107,Y)
880 NEXT Y
890 COLOR 1
900 B=69:C=91
905 FOR Y=33 TO 12 STEP -1
910 B=B+.5:C=C-.5
920 FOR X= B TO C
930 SET(X,Y)
940 SET(X+27,Y)
950 NEXT X:NEXT Y
1000 GOTO 1000
2000 BSAVE"HOUSE",7000,77FF
```

[illegible]

VZ USER   MARK HARWOOD   P.O.BOX 154   DURAL N.S.W.   2158

LE'VZ OOP   J.C.E. D'ALTON   39 AGNES St. TOOWONG QLND. 4066

VZ DOWN UNDER C/- GEORGE SEGGIE P.O.BOX 316 ST.KILDA VIC. 3182

VZ-LINK NEWSLETTER   P.O.BOX 1972   C.P.O. AUCKLAND NEW ZEALAND

VZ USERS GROUP   P.O.BOX 22-094   CHRISTCHURCH NEW ZEALAND

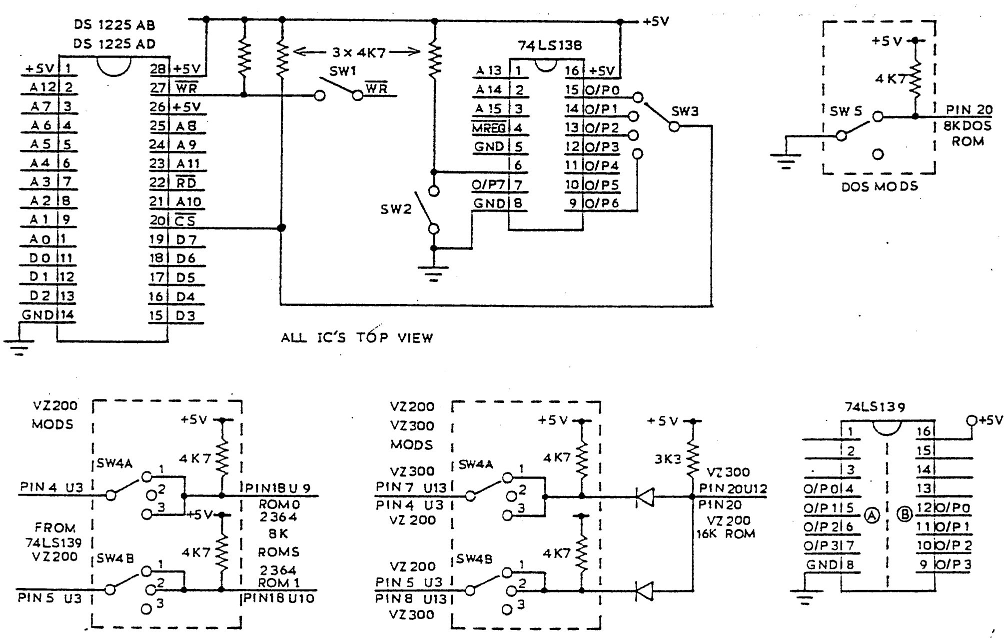
HUNTER VALLEY VZ USERS' GROUP - P.O. BOX 161 JESMOND N.S.W. 2299
EDITOR-JOE LEON (049)51 2756 - SECRETARY-ROSS WOODS (049)71 2843

SUBSCRIPTION - H.V.VZ.JOURNAL - 6 MONTHS $9.00 - 12 MONTHS $18.00
New Zealand - 6 MONTHS $12.00 - 12 MONTHS $24.00

MEETINGS - 2nd. FRIDAY of MONTH at NEW LAMBTON COMMUNITY CENTRE
CNR. ALMA RD. and CROMWELL ST. NEW LAMBTON - Upstairs Hall.

NOTE :- When writing to any above or H.V.VZ. Users' Group for information please enclose a S.S.A.E. or 2 Int. Reply Coupons.

8K BUILT IN BATT. RAM BY JOE LEON



The 2K ZERO Power Ram featured in issue #12 has it's uses. It fills up the 2K gap nicely left by the DOS. One of it's limitations is the small memory size available. This article deals with an 8K version. Like the 2K one the 8K Ram also has Built In LITHIUM Batteries and will be refered to as 8K BIB (Built In Batteries) Ram. Data rentention in the absence of power is over 10 years after which it behaves like a normal CMOS Ram.

Non Disk Drive owners will most likely put the 2 Rams in the 10K Cart. area. Although the 8K BIB Ram can be used there it can be put to more productive work.

This 8K BIB Ram will prove to be most useful in activating all the hidden commands/functions in the Basic Roms or enhancing the DOS. Imagine being able to change at will any command or routine or replace them with your own. The Basic and DOS Roms are full of bugs which persons who know how will be able to fix.

Before proceeding with the article I would like to point out that you'll need the Technical Reference Manual for reference and trouble shooting. And now to the project. There are five circuits although two are very similiar and we'll deal with each one in turn.

8K BIB RAM CIRCUIT :-

The 74LS138 is configured to decode in 8K blocks. Using SW3 (Switch 3) you can place the 8K BIB Ram in the following memory locations. 0000H-1FFFH (Rom 0), 2000H-3FFFH (Rom 1), 4000H-5FFFH (Dos Rom) and at C000H-DFFFH.

If you have a VZ200 then you may have to select a lower O/P than O/P 6. the idea of having the 8K BIB Ram at higher memory is to be able to transfer the contents of one of the 3 lower 8K blocks for modification. Then when it's done, switch out one of the 8K blocks and switch in the 8K BIB Ram in it's place for testing.

Switch 1 is the WRITE ENABLE/DISABLE Switch. Close to WRITE, Open to WRITE PROTECT.

Switch 2 is optional. It allows you to disable the 8K BIB Ram. Close to disable, Open to enable.

VZ200 MODS CIRCUIT :-

MOST VZ200's have 2X8K Basic Roms. These will have to be disabled or enabled in turn as shown in the diagram. For both Roms the CS (Chip Select) line must be cut and the circuit within the broken line box inserted. Switch 4 is Double Pole Triple Throw type.

Switch position 1 - Rom 0 and Rom 1 are enabled.
Switch position 2 - Rom 0 is disabled and Rom 1 is still enabled.
Switch position 3 - Rom 0 is re-enabled while Rom 1 is disabled.

NOTE :- Any switching must be done with the power off except for Switch 1 which can be used any time.

VZ 200/300 MODS :-

This circuit is for VZ 200/300 with 16K Roms. The decoder (U3-74LS139) in VZ200 and (U13) in VZ300 send out Rom 0 and Rom 1 CS signals. They are ANDed together by the DIODE AND GATE which is fed into the CS input pin of the 16K Rom. The same switch arrangement as with the previous mod. applies. No tracks have to be cut as it is easier to unsolder one end of the DIODE and make your connections to that end making sure it is the correct end. In the vacated hole from the DIODE you can put in the switch lead.

DOS MOD :-

This is even simpler to achieve. Cut the track between pin 20 of the 8K Rom and Ground and insert Resistor and Switch as shown in the circuit.

U3 - 74LS139 VZ200 DECODER :-

Unlike the 74LS138 this is a Dual Decoder. I labelled each half 'A' and 'B' respectively. Side 'A' is configured to decode in 8K blocks while side 'B' decodes in 2K blocks.

Before proceeding further a bit of an explanation on decoding and CS (Chip Select) about which I had some enquiries. I'll use the 74LS138 as an example.

The INPUTS decide what size blocks it will decode in. Starting with O/P 0 to O/P 7 each O/P goes to zero volts in turn activating each Ram/Rom the O/P's are connected to. Only one Ram/Rom can be active at any one time. The CS select pin has to be taken to zero volts to activate the particular Ram/Rom. So inversely if we want to deactivate a Ram/Rom we can use a pull up resitor to +5V as shown in the circuits.

In the tables below the addresses are given for each decoded block for both the 74LS138 and 74LS139 decoders.

74LS138 decoder
O/P 0 - 0000H-1FFFH (ROM 1)
O/P 1 - 2000H-3FFFH (ROM 2)
O/P 2 - 4000H-5FFFH (DOS ROM)
O/P 3 - 6000H-7FFFH
O/P 4 - 8000H-9FFFH
O/P 5 - A000H-BFFFH
O/P 6 - C000H-DFFFH
O/P 7 - E000H-FFFFH

74LS139 decoder
O/P 0 - 0000H-1FFFH (ROM 1)
O/P 1 - 2000H-3FFFH (ROM 2)
O/P 2 - 4000H-5FFFH (DOS ROM)
O/P 3 = 6000H-7FFFH - Side A ↑
O/P 0 - 6000H-67FFH - Side B
O/P 1 - 6800H-6FFFH - Side B
O/P 2 - 7000H-77FFH - Side B
O/P 3 - 7800H-7FFFH - Side B

The 74LS139 in the VZ200 has two unused outputs of particular interest to us.

O/P 1 - Pin 6, side A - 4000H-5FFFH (8K Block)
O/P 2 - Pin 12, side B - 6000H-67FFH (2K Block)

You'll notice that's the 10K Cartidge area. I used the 2K decoded O/P for the 2K Zero Power Ram as described in last issue which saved me building a decoder. I'm lazy, if it's already there I'll use. Why reinvent the wheel. Happy hacking.

```
0 '*********************************************************
2 '*  CONVERT DEC TO HEX TO BINARY TO LO-BYTE AND HI-BYTE  *
4 '*  MODIFIED AND ENHANCED FOR THE VZ  BY  DAVE MITCHELL  *
6 '*********************************************************
8 :
10 POKE30862,0:POKE30863,114:POKE29184,243:POKE29185,201
20 POKE30777,1:Z$="                                "
30 CLS:PRINT" THIS UTILITY WILL DISPLAY THE   EQUIVALENT";
40 PRINT" VALUES OF DECIMAL,   HEX AND BINARY ON THE SCREEN"
50 PRINT" AND TO THE PRINTER."
60 PRINT@164,"[illegible] ENTER DECIMAL VALUE"
70 PRINT@228,"[illegible] ENTER BINARY  VALUE"
80 PRINT@292,"[illegible] ENTER HEX     VALUE"
90 PRINT@356,"[illegible] ENTER L.S.B.& M.S.B."
100 PRINT@420,"[illegible]"
110 AA$=INKEY$:A$=INKEY$:IFA$=""THEN110
120 SOUND30,1
130 IFA$="A"THENGOSUB180:GOSUB230:GOTO110
140 IFA$="B"THENGOSUB180:GOSUB240:GOTO110
150 IFA$="C"THENGOSUB180:GOSUB250:GOTO110
160 IFA$="D"THENGOSUB180:GOSUB720
170 GOTO110
180 PRINT@0,Z$:PRINT@32,Z$:PRINT@64,Z$:PRINT@96,Z$
190 PRINT@0,"DECIMAL =":PRINT@32,"HEX     ="
200 PRINT@64,"BINARY  =":PRINT@96,"L.S.B   ="
210 PRINT@113,"M.S.B. =":RETURN
220 PRINT@448,Z$:RETURN
230 GOSUB410:GOSUB540:GOSUB260:RETURN
240 GOSUB470:GOSUB540:GOSUB260:RETURN
250 GOSUB620:GOSUB420:GOSUB260:RETURN
260 PRINT@480,"PRINT OUT [illegible]";
270 AA$=INKEY$:A$=INKEY$:IFA$<>"Y"ANDA$<>"N"THEN270
280 SOUND30,1
290 PRINT@480,"                   ";
300 IFA$="N"THEN400
310 IFC<=32767THEN330
320 LPRINT"DECIMAL VALUE ="C;"   ( ";INT(C-65536);")":GOTO340
330 LPRINT"DECIMAL VALUE ="C
340 LPRINT" HEX     VALUE = ";H1$
350 LPRINT" BINARY VALUE = ";C$
360 LPRINT"LEAST SIGNIFICANT BIT=";E2
370 LPRINT"MOST SIGNIFICANT BIT =";E1
380 REM LINE 309 LPRINTSTRING$(20,46)
390 LPRINTSTRING$(20,46)
400 C$="":H1$="":RETURN
410 B$="":PRINT@448,"ENTER DECIMAL VALUE";:INPUTD:C=D:GOSUB220
420 X=USR(0):B=D/2:D=INT(B):IFB=DTHENB$="0"+B$ELSEB$="1"+B$
430 IFD>0THEN420ELSEPRINT@74,B$
440 C$=B$:B$=""
450 PRINT@9,C:IFC<=32767THENRETURN
460 PRINT@20,CHR$(104);" ";C-65536;CHR$(105):RETURN
470 D=0:PRINT@448,"ENTER BINARY ";:INPUTB$:C$=B$:GOSUB220
480 X=USR(0)
490 FORI=LEN(B$)TO1STEP-1:IFMID$(B$,I,1)="1"THEND=D+2^(LEN(B$)-I)
500 NEXT:PRINT@74,C$
510 IFD<=32767THEN530
520 PRINT@20,CHR$(104);" ";INT(D-65536);CHR$(105)
530 PRINT@9,D:C=D:D=0:B$="":I=0:RETURN
540 X=USR(0):AD=C
550 B=INT(AD/256):E1=INT(B):GOSUB580:H$=B$
560 B=AD-(256*B):E2=INT(B):GOSUB580:H$=H$+B$
570 PRINT@42,H$:H1$=H$:PRINT@105,E2:PRINT@121,E1:RETURN
```

```
580 H=INT(B/16):L=B-(H*16):B$=""
590 D=H:GOSUB600:D=L:GOSUB600:RETURN
600 IFD>9THENB$=B$+CHR$(55+D)ELSEB$=B$+CHR$(48+D)
610 RETURN
620 PRINT@448,"ENTER HEX EG. 0000 ";:INPUTH$:H1$=H$:GOSUB220
630 X=USR(0):H$=RIGHT$(H$,4)
640 HH$=LEFT$(H$,2):GOSUB680:E=H*256:E1=H
650 HH$=RIGHT$(H$,2):GOSUB680:E=E+H:E2=H
660 PRINT@42,H1$:D=E:C=E:B$="":PRINT@105,E2:PRINT@121,E1
670 RETURN
680 H=0:A$=LEFT$(HH$,1):GOSUB700:H=D*16
690 A$=RIGHT$(HH$,1):GOSUB700:H=H+D:RETURN
700 D=ASC(A$):IFD<58THEND=D-48ELSED=D-55
710 RETURN
720 B$="":PRINT@448,"ENTER LSB.";:INPUTS:GOSUB220
730 PRINT@448,"ENTER MSB.";:INPUTM:GOSUB220:X=USR(0)
740 D=S+256*M:C=D:S=0:M=0:GOSUB420:GOSUB540:GOSUB260:RETURN
```

VZ MODS INFORMATION BY ROSS WOODS

MODIFYING VZ 200 / 300 16 K RAM EXPANSION

Use of the VZ200 RAM PACK will give only 6K EXTRA to the VZ 300 COMPUTER. To get the FULL 16K with VZ300 we refer you to an article by STEVE OLNEY in ETI, FEBRUARY 1986, (Page 71-74). The modification involves "PIGGYBACKING" of an extra IC, a (74LS00) and cutting of ONE PCB TRACK. (It is possible to fit a SWITCH SO THAT IT CAN BE USED WITH BOTH VZ 200 / 300 COMPUTERS).

On the other hand you may want to use a VZ300 RAM PACK with a VZ200 COMPUTER, the details of this modification is normally supplied together with the instructions for the RAM PACK from DICK SMITH ELECTRONICS. It involves moving LINKS on the PCB. (It may be possible to fit a SWITCH SO THAT IT COULD BE USED WITH BOTH VZ 200 / 300 COMPUTERS).

INCREASING VZ 200 / 300 INTERNAL MEMORY

SUPER II VZ200 MODIFICATION - ETI PROJECT 687

The SUPER II 34K MEMORY BOARD REPLACES THE VZ 200 INTERNAL 6K MEMORY BOARD. The new Top of MEMORY now is 'FFFF'. This Project was PUBLISHED in ETI JULY 1986, Pages 55-60 by MATTHEW SORREL. It uses 4 x 8K & 1 x 2K MEMORY chips on a PCB REPLACING THE VZ200 INTERNAL 6K MEMORY PCB.

FOR those who have a VZ 300 and require FULL EXPANSION to Top of MEMORY 'FFFF' there was an article in ISSUE Number 3, Page 12-13 by Dave Boyce in the Hunter Valley VZ Users' Group Newsletter (Now known as HUNTER VALLEY VZ JOURNAL).

Write to Dave BOYCE or H.V.VZ Users' group for more information.

ENQUIRIES RE ETI BACK ISSUES OR PHOTOSTATS OF ARTICLES WRITE TO
ETI READER SERVICES, 180 BOURKE RD. ALEXANDER N.S.W. 2015.
OR PO BOX 227 WATERLOO, N.S.W. 2017.

COST OF BACK ISSUES $4.00 IF AVAILABLE.
COST OF PHOTOSTATS $4.00 (ARTICLE IN 1 ISSUE).
COST OF PHOTOSTATS $8.00 (ARTICLE IN 2 OR MORE ISSUES).

One source for ETI 687 P.C.B. Supplies is :-
R.C.S. RADIO PTY. LTD. Phone (02) 587 3491
651 FOREST ROAD BEXLEY 2207 AUSTRALIA

Most of the VZ's BASIC words that were never enabled in the VZ have dedicated processing routines in ROM. Each of these words is listed along with its token code, its start address in the WORD TABLE, and the start address for the routine that processes it. The routine addresses are listed in decimal, then in LO HI format (where address = LO+HI *256) and in hex.

NOTE :- Continued on next page . . .

TOKEN ADD.	TOKEN	WORD	ROUTINE DEC	LO	HI	ADDRESS HEX
5712	128	END	7598	174	29	1DAE
5715	129	FOR	7329	161	28	1CA1
5718	130	RESET	312	56	1	0138
5723	131	SET	309	53	1	0135
5726	132	CLS	457	201	1	01C9
5732	134	RANDOM	467	211	1	01D3
5738	135	NEXT	8886	182	34	22B6
5742	136	DATA	7941	5	31	1F05
5746	137	INPUT	8602	154	33	219A
5751	138	DIM	9736	8	38	2608
5754	139	READ	8687	239	33	21EF
5758	140	LET	7969	33	31	1F21
5761	141	GOTO	7874	194	30	1EC2
5765	142	RUN	7843	163	30	1EA3
5768	143	IF	8249	57	32	2039
5770	144	RESTORE	7569	145	29	1D91
5777	145	GOSUB	7857	177	30	1EB1
5782	146	RETURN	7902	222	30	1EDE
5788	147	REM	7943	7	31	1F07
5791	148	STOP	7953	169	29	1DA9
5795	149	ELSE	7943	7	31	1F07
5799	150	COPY	14610	18	57	3912
5803	151	COLOR	14493	157	56	389D
5808	152	VERIFY	14136	56	55	3738
5814	153	DEFINT	7683	3	30	1E03
5820	154	DEFSNG	7686	6	30	1E06
5826	155	DEFDBL	7689	9	30	1E09
5832	156	CRUN	14126	46	55	372E
5836	157	MODE	11875	99	46	2E63
5840	158	SOUND	11253	245	43	2BF5
5845	159	RESUME	8111	175	31	1FAF
5851	160	OUT	11003	251	42	2AFB
5854	161	ON	8044	108	31	1F6C
5911	175	LPRINT	8295	103	32	2067
5920	177	POKE	11441	177	44	2CB1
5924	178	PRINT	8303	111	32	206F
5929	179	CONT	7652	228	29	1DE4
5933	180	LIST	11054	46	43	2B2E
5937	181	LLIST	11049	41	43	2B29
5942	182	DELETE	11206	198	43	2BC6
5948	183	AUTO	8200	8	32	2008
5952	184	CLEAR	7802	122	30	1E7A
5957	185	CLOAD	13910	86	54	3656
5962	186	CSAVE	13481	169	52	34A9
5967	187	NEW	6985	73	27	1B49
5970	188	TAB(	NO	ADDRESS		LIST
5974	189	TO	NO	ADDRESS		LIST
5978	191	USING	NO	ADDRESS		LIST

TOKEN ADD.	TOKEN	WORD	ROUTINE DEC	LO	HI	ADDRESS HEX
5983	192	VARPTR	NO	ADDRESS		LIST
5989	193	USR	NO	ADDRESS		LIST
5992	194	ERL	NO	ADDRESS		LIST
5995	195	ERR	NO	ADDRESS		LIST
5998	196	STRING$	NO	ADDRESS		LIST
6010	198	POINT	NO	ADDRESS		LIST
6020	200	MEM	NO	ADDRESS		LIST
6023	201	INKEY$	NO	ADDRESS		LIST
6029	202	THEN	NO	ADDRESS		LIST
6033	203	NOT	NO	ADDRESS		LIST
6036	204	STEP	NO	ADDRESS		LIST
6040	205	+	NO	ADDRESS		LIST
6041	206	+	NO	ADDRESS		LIST
6042	207	*	NO	ADDRESS		LIST
6043	208	/	NO	ADDRESS		LIST
6044	209	^	NO	ADDRESS		LIST
6045	210	AND	NO	ADDRESS		LIST
6048	211	OR	NO	ADDRESS		LIST
6050	212	>	NO	ADDRESS		LIST
6051	213	=	NO	ADDRESS		LIST
6052	214	<	NO	ADDRESS		LIST
6053	215	SGN	2442	138	9	098A
6056	216	INT	2871	55	11	0B37
6059	217	ABS	2423	119	9	0997
6062	218	FRE	10196	212	39	27D4
6065	219	INP	10991	239	42	2AEF
6068	220	POS	10229	245	39	27F5
6071	221	SQR	5095	231	19	13E7
6074	222	RND	5321	201	20	14C9
6077	223	LOG	2057	9	8	0809
6080	224	EXP	5177	57	20	1439
6083	225	COS	5441	65	21	1541
6086	226	SIN	5447	71	21	1547
6089	227	TAN	5544	168	21	15A8
6092	228	ATN	5565	189	21	15BD
6095	229	PEEK	11434	170	44	2CAA
6129	239	CINT	2687	127	10	0A7F
6133	240	CSNG	2737	177	10	0AB1
6137	241	CDBL	2779	219	10	0ADB
6141	242	FIX	2854	38	11	0B26
6144	243	LEN	10755	3	42	2A03
6147	244	STR$	10294	54	40	2836
6151	245	VAL	10949	197	42	2AC5
6154	246	ASC	10767	15	42	2A0F
6157	247	CHR$	10783	31	42	2A1F
6161	248	LEFT$	10849	97	42	2A61
6166	249	RIGHT$	10897	145	42	2A91
6172	250	MID$	10906	154	42	2A9A

The WORD TABLE divides into three parts.

Words from END (token 128) to NEW (token 187) are all words that must begin a BASIC statement if they are used at all. Some (such as END, CLS, NEW, STOP) must stand alone as the entire BASIC statement. Others (NEXT, PRINT, RUN, etc.) may stand alone as an entire statement or be the primary word of an extended statement. Each of these words has a two byte pointer in the POINTER TABLE that follows the WORD TABLE. The pointer contains the address (in LO HI format) of the routine that executes the word (don't worry, its quite official--our VZ is no murderer).

Words from SGN (token 215) to MID$ (token 250) are all words that cannot occur at the start of a BASIC statement. Each has a pointer in the POINTER TABLE which precedes the WORD TABLE, where again the pointer addresses a processing routine (execute!) for that word.

Words from TAB((token 188) to < (token 214) do not have pointers in either of the POINTER TABLES. They are executed by the primary words of statements in which they occur. Strictly speaking it is the tokens that do all the dirty work. The words are just there for the convenience of the human operator. imagine trying to figure out a program listing which only displayed tokens.

5640 to 5711: ROUTINE POINTER TABLE 1
5712 to 6177: WORD TABLE
6178 to 6297: ROUTINE POINTER TABLE 2

NOTES ON PROGRAM WORDS FROM LAST ISSUE :-

When the program is RUN, tokens 205 to 208 and 212 to 214 designate words that display on screen as graphic characters. The program is accurate. The first byte of each word being in inverse form to mark the start of a new word. This is strictly true for words which begin with alphabetical characters, the majority, but not for those few words that consist of nonalphabetical characters. A more accurate wording would be:-

"The first byte of each word being the ASCII character code + 128 to mark the start of a new word." For alphabetical characters, adding 128 to the ASCII code will yield the inverse character code. Most BASIC words begin with an alphabetical character, so most of the the words in the WORD TABLE will begin with an inverse character. However, a few BASIC words consist of a single nonalphabetical character: +,-,*,/,^,>,=,<

They too are represented in the WORD TABLE by the ASCII code + 128, but this does not yield the inverse character codes. On the VZ, for those characters with ASCII codes less than 64, the inverse character code is the ASCII code + 192. Still, when the VZ BASIC INTERPRETER comes across one of these words in the WORD TABLE, it subtracts 128 to get the ASCII code.

Add this line to the WORD program to get it to identify the few BASIC words that consist of single, nonalphabetical characters:-

```
38 IFP<192ANDP>159THEN PRINT"WORD IS ";CHR$(P-128);
```

NOTE :- See next page for printout using the added line 38.

SHOULD START WITH TOKE 190

```
5712 128:197■ 78N 68D
5715 129:198■ 79O 82R
5718 130:210■ 69E 83S 69E 84T
5723 131:211■ 69E 84T
5726 132:195■ 76L 83S
5729 133:129 ■  0 0
5732 134:129 ■  0 0 0 0 0
5738 135:206■ 69E 88X 84T
5742 136:196■ 65A 84T 65A
5746 137:201■ 78N 80P 85U 84T
5751 138:196■ 73I 77M
5754 139:210■ 69E 65A 68D
5758 140:204■ 69E 84T
5761 141:199■ 79O 84T 79O
5765 142:210■ 85U 78N
5768 143:201■ 70F
5770 144:210■ 69E 83S 84T 79O 82R 69E
5777 145:199■ 79O 83S 85U 66B
5782 146:210■ 69E 84T 85U 82R 78N
5788 147:210■ 69E 77M
5791 148:211■ 84T 79O 80P
5795 149:197■ 76L 83S 69E
5799 150:195■ 79O 80P 89Y
5803 151:195■ 79O 76L 79O 82R
5808 152:214■ 69E 82R 73I 70F 89Y
5814 153:129 ■  0 0 0 0 0
5820 154:129 ■  0 0 0 0 0
5826 155:129 ■  0 0 0 0 0
5832 156:195■ 82R 85U 78N
5836 157:205■ 79O 68D 69E
5840 158:211■ 79O 85U 78N 68D
5845 159:129 ■  0 0 0 0 0
5851 160:207■ 85U 84T
5854 161:129 ■  0
5856 162:129 ■  0 0 0
5860 163:129 ■  0 0 0 0
5865 164:129 ■  0 0
5868 165:129 ■  0 0
5871 166:129 ■  0 0 0 0
5876 167:129 ■  0 0 0
5880 168:129 ■  0 0 0 0
5885 169:129 ■  0 0 0
5889 170:129 ■  0 0 0
5893 171:129 ■  0 0 0
5897 172:129 ■  0 0 0
5901 173:129 ■  0 0 0
5905 174:129 ■  0 0 0 0 0
5911 175:204■ 80P 82R 73I 78N 84T
5917 176:129 ■  0 0
5920 177:208■ 79O 75K 69E
5924 178:208■ 82R 73I 78N 84T
5929 179:195■ 79O 78N 84T
5933 180:204■ 73I 83S 84T
5937 181:204■ 76L 73I 83S 84T
5942 182:129 ■  0 0 0 0 0
5948 183:129 ■  0 0 0
5952 184:195■ 76L 69E 65A 82R
5957 185:195■ 76L 79O 65A 68D
5962 186:195■ 83S 65A 86V 69E
5967 187:206■ 69E 87W
5970 188:212■ 65A 66B 40(
5974 189:212■ 79O

5976 128:129 ■  0
5978 129:213■ 83S 73I 78N 71G
5983 130:129 ■  0 0 0 0 0
5989 131:213■ 83S 82R
5992 132:129 ■  0 0
5995 133:129 ■  0 0
5998 134:129 ■  0 0 0 0 0 0
6005 135:129 ■  0 0 0 0
6010 136:208■ 79O 73I 78N 84T
6015 137:129 ■  0 0 0 0
6020 138:129 ■  0 0
6023 139:201■ 78N 75K 69E 89Y 36$
6029 140:212■ 72H 69E 78N
6033 141:206■ 79O 84T
6036 142:211■ 84T 69E 80P
6040 143:171■ WORD IS +
6041 144:173■ WORD IS -
6042 145:170■ WORD IS *
6043 146:175■ WORD IS /
6044 147:222■
6045 148:193■ 78N 68D
6048 149:207■ 82R
6050 150:190■ WORD IS >
6051 151:189■ WORD IS =
6052 152:188■ WORD IS <
6053 153:211■ 71G 78N
6056 154:201■ 78N 84T
6059 155:193■ 66B 83S
6062 156:129 ■  0 0
6065 157:201■ 78N 80P
6068 158:129 ■  0 0
6071 159:211■ 81Q 82R
6074 160:210■ 78N 68D
6077 161:204■ 79O 71G
6080 162:197■ 88X 80P
6083 163:195■ 79O 83S
6086 164:211■ 73I 78N
6089 165:212■ 65A 78N
6092 166:193■ 84T 78N
6095 167:208■ 69E 69E 75K
6099 168:129 ■  0 0
6102 169:129 ■  0 0
6105 170:129 ■  0 0
6108 171:129 ■  0 0
6111 172:129 ■  0 0
6114 173:129 ■  0 0
6117 174:129 ■  0 0 0
6121 175:129 ■  0 0 0
6125 176:129 ■  0 0 0
6129 177:129 ■  0 0 0
6133 178:129 ■  0 0 0
6137 179:129 ■  0 0 0
6141 180:129 ■  0 0
6144 181:204■ 69E 78N
6147 182:211■ 84T 82R 36$
6151 183:214■ 65A 76L
6154 184:193■ 83S 67C
6157 185:195■ 72H 82R 36$
6161 186:204■ 69E 70F 84T 36$
6166 187:210■ 73I 71G 72H 84T 36$
6172 188:205■ 73I 68D 36$
```

D.S.E. GAME - GALAXON - CAT #X-7352

Well D.S.E. have done it again, another excellent VZ game. I wonder how long they have been sitting on this beauty ?

Anyway, let me tell you about Galaxon :-

You must fire at a pack of aliens above you, while trying not to be hit by the diving aliens yourself.

These aliens dive on you from the pack and try to crash into you. To make matters even worse, these diving aliens drop bombs that will surely kill you if you don't avoid them.

In the attacking pack there are three different types of alien and each type is worth a different score.

Also there are more points awarded for hitting the aliens while they are diving rather than hitting them while they are just sitting still in the pack.

If after all this, you manage to avoid the aliens, they reform in the pack at the top of the screen. You get three lives and also get an extra life for every 10,000 points you score.

You would really have to see and play this game to really appreciate how good it really is. It's not an easy game to play, and who would want it to be.

I liked this game from the moment it started loading. First you get a Galaxon title page, followed by an instruction page, which you have a chance to read while the game is loading.

Key control is excellent, as is the joystick option. One thing I felt was a good idea, was using the (J) or (K) key option to start each new game instead of the (S) key. That way if you get sick of using your joysticks and feel like a change, you don't have to switch off and reload to use the keys. (REMEMBER Don't plug in or unplug your joysticks without first TURNING off your VZ, you will do damage to your system if you do).

Galaxon is like a vz invaders game when you first get started, but are you in for a surprise when these aliens really start moving.

The game is fast moving and excellent use is made of the VZ's graphics and sound effects.

I had trouble turning off my computer once I had started playing this game, and when this happens you know you've got a winner. The programmer "Stephen Clarke" should write and market his own software, if this game is anything to go by.

Anyway, "Galaxon" must follow as a very close second, when it comes to super VZ programs, just behind "Dawn Patrol". D.S.E. haven't had many programs as good as this one, which is a shame, because we users know what the VZ is capable of. Don't we !!

By Peter J. Hill (Auckland N.Z.)

Now that You have made Your AEM4505 SPEECH SYNTHESISER, let's add what could be best described as a TONE control. Before continuing with the construction changes, a word of WARNING.

The two Tone controls should be left in their central positions and only moved minimum amounts to suit ones' own taste. Move the controls TOO far and the Speech Synth will go into Oscilation, if the RESET button doesn't regain control then powering down of the Speech Synth (only) will be necaessry.

YOU have been WARNED.

Construction Notes :-.

1> Using a small piece of Vero-board, construct the circuit diagram as shown. I suggest that sockets be installed under the two Ic's. Although the diagram looks involved, there are very few wires to connect to the Main 4505 P.C.B. These being :-

GND - 0 Volts
+ 5 Volts
1 wire from the output of the 74LS00 gate to the SPO chip plus
2 wires to the PITCH control and
3 wires to the RANGE control from the small board to the pots.

2> Remove capacitors C4 and C5.

3> Remove X-TAL2 , it's next to the SPO chip.

Reconnecting Notes :-

4> Solder the GND (0 V) wire to the 4505 P.C.B. followed by the +5 Volt wire.

5> Next - solder the wire marked TP on the diagram below into the hole nearest the SPO chip left vacant by the X-TAL. This hole is connected on the P.C.Board to pin 27 of the SPO chip.

6> Connect the remaining wires to their respective control pots.

7> On the BACK of each Control Pot, assuming they are metal, scratch the surface until you have a clean surface. Tin each of these bare spots. From the GND terminal of either the VOL or RANGE pot, run a wire to the back of each pot and solder to the 'Tinned Bare Spots'. This will stop (hopefully) the pots acting as antennas when you touch them - honest.

8> Reconnect any other wires that may have been disconnected when you started on the job.

9> Set the two new pots to their central positions.

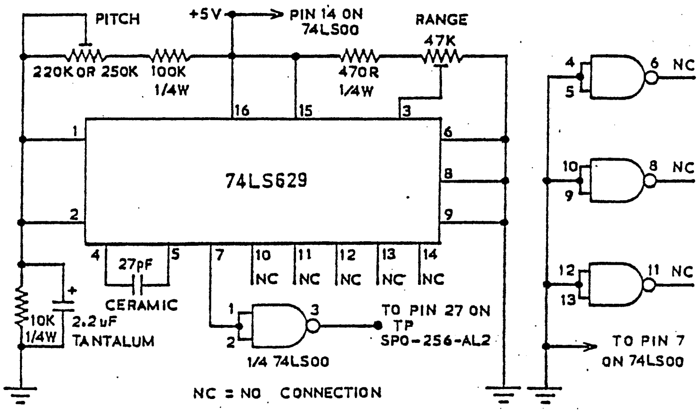
O.K. Connect the Speech Synth back onto the VZ and turn on. Everything alright ? Did you get the familiar ' OK ' from the Speech Synth ? If not then turn off and check everything from the start.

Load in and RUN "TALK A" and check that the two controls are working.

NOTE :- Circuit Diagram and Parts List on next page . . .

Parts List for the Tone Control for AEM 4505 Speech Synth. - Dick Smith catalogue numbers given where possible.

1 - 74LS629 - ALTRONICS Z 8429
1 - 74LS00 - DSE Z-4900
1 - 470R resistor - DSE R-0566
1 - 10K resistor - DSE R-0598
1 - 100K resistor - DSE R-0624
1 - 27pF ceramic capacitor - DSE R-2245
    See note below.
1 - 2u2 35 volt Tantalum capacitor - DSE
    R-4730
1 - 47K or 50K potentiometer - DSE R-1809
1 - 220K potentiometer. Yes Virginia ,
    they do exist.
1 - 14 PIN DIP socket - DSE P-4140
1 - 16 PIN DIP socket - DSE P-4160



Note on 27pf cap.:- The value of this cap. may have to be varied from anywhere between 18pf and 33pf. A trimmer cap. from D.Smiths can also be used in place of the fixed value cap. D.S. No# - R2930. It is possible to mount the small board on the back of one of the control pots, this will save using long wires to that pot, PLUS, it saves having to find a mounting place for the small board.

I have made this modification to both my original Speech Synth and my (home made version of the) AEM4505, in either case, the mod works quite well.

Cost - $12 to $15 - Happy Hackin'. Dave Boyce June '87

Routine to make the unit Talk.

```
100 A$="HELLO-THERE"
110 REM WHERE A$ is to be output to the Speech Synthsiser
120 GOSUB 15020 :REM OR GOTO
```

This is the sub-routine to be used :-

```
15010 REM - IN USE -> A$, A and T
15020 FOR T=1 TO LEN(A$)
15030 A=ASC(MID$(A$,T,1))
15040 OUT12,A :REM OUT13,0 MAY BE NECESSARY AS WELL
15050 NEXT
15060 LPRINT:LPRINT:REM THESE MUST BE HERE
15070 RETURN :REM OR GOTO 0000 - CONTINUE ON
```

Make A$ equal to any String to be output to the Speech Synth.
USE HYPHENS between words, this helps the Speech flow evenly.
2 Hyphens (or more) will slow the Speech down.
Commas have a pausing effect as well.
A (.) Period will make the Speech Pause between words (long pause). 2 Periods (..) will pause even longer.

EXAMPLES :-

```
10 A$ ="DOGGY.WFFWFF-WFF-WFFWFF-RFF"
20 A$= "STEAM-TRAIN.SHSHSHSHSHSS,TOOT-TOOT,SHSHSH"
30 A$= "WHISPERING.SHSH.SH..SH-SH-SH-SHSHSH-SH-SHSHSH"
40 A$= "MACHINE-GUN.GGGGGGGGGGGGGGGGGGGGGGGGGGGG":REM (28G's)
```

DOT MATRIX PRINTERS by Larry Taylor

The VZ is a versatile, little computer, especially if teamed with a peripheral device. When that device is an EPSON type dot matrix printer, its capabilities can greatly enhance the VZ's versatility.

Dot matrix printers form their characters from a series of dots. This type of printer falls into one of two categories either impact or non-impact. There are three main types of non-impact printers, thermal, electrostatic and ink jet. Thermal print heads contain a set of fixed pins, which are heated rapidly to produce a dot on specially treated, heat sensitive paper. Electrostatic print heads, also consist of fixed pins, which produce a tiny electrical discharge between the pins and an aluminium coated paper, forming the dot. Ink jet print heads are comprised of a vertical line of pin holes, through which a fine jet of ink is squirted onto the paper to create each dot.

Currently, impact type printers are the most common. The print head in an impact type printer usually consists of a line of pins arranged vertically. These pins strike against an inked ribbon to leave an impression on the paper. Each pin can be fired independantly. By firing all or some of these pins, a column of dots can be formed. The matrix used for most upper case (capital) letters is 5 dots wide and 7 dots high. To produce a character, the pins are fired forming a column of dots. The print head then moves one dot position and the pins fire again. This happens three more times to produce an upper case letter 5 dots wide.

Unfortunately, while the principle of their operation is similar, dot matrix printers differ in other ways. Software codes that are sent from the computer to the printer determine which pins will be fired. Not all printers will react to those codes in the same way. There are basically three families, EPSON, TANDY and APPLE, each conforming to a different standard. Both the EPSON and APPLE types address 8 pin print heads, whilst the TANDY types possess only 7 pins. Each pin is given a weighted value. In the case of EPSON types, the topmost pin has a value of 128, whilst the bottom pin has the value 1. These values are reversed for APPLE and TANDY printers. The bottom pin on a TANDY type is the seventh one and it has the value 64. TANDY type printers will only accept codes in the range from 1 to 127. The GP-100 belongs to the TANDY family and the VZ's ROM is set up to work with this type of printer. Since the codes used aren't compatible with EPSON type printers, a software patch is needed. The value of such a patch can be determined by looking at the relative merits of the two types of printers.

The GP-100 has tractor feed only, which requires the use of paper with sprocket holes. It has a print speed of 30 characters per second (30 cps) and is capable of unidirectional (left to right) printing only. This printer has both character and graphics modes. In character mode it possesses just the one typeface, which has no true descenders. That is, the tails on the lower case letters g,j,p,q and y do not descend below the line as they normally would. The only typestyle change, that can be made to the existing typeface, is that each character can be printed in double width (expanded) form.

On the other hand, software such as the VZ-EPSON Printer Patch, WORDPRO and QUICKWRITE, allows the VZ to make the most of the facilities provided by EPSON type printers. All allow EPSON codes to be sent directly to the printer.

Consequently, instead of being limited to the GP-100's meagre repertoire, a whole range of printers, with a host of enhanced features, is on offer. Most provide tractor and friction paper feeds, though a few require that the tractor feed be bought separately. In some cases, it may be possible to purchase a cut paper feeder, which can be useful, when printing multiple copies of correspondence on single sheets. Logic seeking (where the head looks to move the shortest distance possible) and bidirectional printing (the head prints on both left to right and right to left passes), means speeds can range from a minimum of 80 cps to 200 cps and more. These speeds are, of course, optimum values.

Though most printers possess only two typefaces (character sets), pica and italic, which are stored in an internal ROM, they compensate for this by offering a variety of typestyles. Different typestyles are formed by slightly modifying each character as it is printed. These include elite, condensed and double width. A few printers also offer double height and reverse print. In addition, there are features such as subscript (below the line) and superscript (above the line), emphasized and double strike. Proportional spacing is also available, which adjusts the distance between letters, according to their width. Some printers may have a high density (near letter quality) facility as well. With this the print head makes a second pass along each line, filling in the gaps between the dots, which form each character. Although slower, it results in better formed characters.

The ability to control line feeds to within 1/216 of an inch is desirable when compared to the GP-100's 1/6 of an inch. This is particularly useful to anyone, who has struggled to get mailing labels to line up on the printer. Some even allow the paper to be fed in reverse as well as forward. (This entire article was written using WORDPRO and printed in a single pass. The two column format was achieved by altering margin settings and using the printer's reverse feed.) The standard features all have recognized EPSON codes, whereas other features and their codes may vary from one printer to the next. The following table displays the codes, in decimal form, along with the resulting print type.

Standard Epson Codes

[27,80]	Normal Pica Print
[27,112,1]	**Proportional Print**
[27,77]	Elite Print
[27,52]	*Italic Print*
[14]	Double Width
[15]	Condensed Print
[27,69]	**Emphasized Print**
[27,71]	Double Strike Print
[27,45,1]	Underlined Print
[27,83,0]	Superscript Print
[27,83,1]	Subscript Print

Non-Standard Codes

[27,97,1]	High Density Print
[27,126,50,1]	REVERSED PRINT
[27,104]	DOUBLE HEIGH

Whilst there is considerable standardization of EPSON print codes, there seems to be very little when it comes to the format used to present them. When examining printer manuals, the presentation of these codes can appear quite confusing. Some examples are given below. Each is supposed to show which codes are needed to switch on Emphasized Print. As shown in the table, this simply involves sending the two values 27 and 69 to the printer.

Printer Manual 1 ESC+E [(45, C5)H, [69,197] D]
Printer Manual 2 ESC E
Printer Manual 3 <ESC> 69 45 .
Printer Manual 4 <ESC> E
Printer Manual 5 [1B]H [45]H or $[27]_{10}$ $[69]_{10}$

If you're confused by the above, don't feel bad, I found it just as difficult to understand? Basically the codes are presented in three ways, ASCII, Decimal (Base 10) and Hexidecimal (Base 16). ESC stands for Escape and represents the Decimal value 27. In Hexidecimal form the value 27 is equal to 1B. If a number has no identifying character or is followed by either a D or 10, then it is considered to be a decimal number. If it is followed by an H or 16, then it is considered to be Hexidecimal. So the three ways of representing the code required to switch an EPSON printer to Emphasized mode are summarised below.

ASCII	ESC	E
Decimal	27	69
Hexidecimal	1B	45

In VZ BASIC it would be accomplished by:

```
LPRINT CHR$(27);"E";
or LPRINT CHR$(27);CHR$(69);
```

Word processing software such as QUICKWRITE and WORDPRO allow print codes to be embedded in the text. Since these programs bypass the VZ's printer driver, any code value used will be sent directly to the printer. The following examples, illustrate how each of these programs enable a change of typeface at the beginning of a line.

QUICKWRITE embeds codes at the start of a line of text. The code must be preceded by a carriage return <CR>.

eg. <CR>[27][52]This is a word processing program.

WORDPRO uses printer control lines which are preceded by a print flag <PF> and end in a carriage return <CR>.

eg. <PF>N=27,52<CR>
This is a word processing program.

The result, when printed from either program is as follows:

This is a word processing program.

Although the manual says otherwise, WORDPRO is able to change typefaces in the middle of a line. The procedure is a little more complicated and involves turning the line feed command on and off. My thanks to John Chapman for supplying the method used.

eg. <PF>F=N<CR> (set line feed off)
This is a<CR>
<PF>N=27,52<CR> (switch italics on)
word processing<CR>
<PF>N=27,53 F=Y<CR> (italics off, line feed on)
program.

The result, when printed is as follows:

This is a *word processing* program.

I've been informed that a recently updated version of QUICKWRITE will also be able to do this.

Whilst both of these approaches are satisfactory, they require the user to be familiar with all of the different print codes. Some word processing packages allow a label to be assigned to each print code, when the program is first set up. These can then be inserted in the text as required without the user having to know the code.

One problem often experienced, when attempting to use the VZ with an EPSON type printer, is that EPSON codes often alternate the values 1 and 0, for switching on and off a particular feature. For instance, the code used to enable and disable underlining [27,45,n] is identical, except for the last value n. Using the value n=1 selects underlining, whilst the value n=0 turns it off. The current VZ printer routine will not allow the value 0 through to the printer. Therefore, switching on underlining will work, but switching it off will not, and the same applies to similarly structured codes. Also affected, is any print code value greater than 127, which will be interpreted by the VZ's ROM routine as being either a graphics or inverse character. Since TANDY type printers do not store any characters in this range, the value will be intercepted by the VZ's ROM and replaced by a stream of graphics data. Whilst this data will enable a printer such as the GP-100, to print the character, it will not be compatible with an EPSON printer. These difficulties can be overcome by using a printer patch or by sending the value directly out the printer ports (0DH and 0EH). Print codes sent to the printer in this way, will be acted on without interference. It should be noted, however, that any characters printed by an EPSON printer, corresponding to codes in the range from 95 to 255, will not be the same as those displayed on the VZ's screen.

A number of printers come equipped with a RAM buffer. Instead of the computer being tied up waiting for the printer, the buffer acts as a storage area for data to be printed and so frees the computer sooner. A second use, is as an area for downloadable characters. These are shapes, which the user may design or which are modifications of the existing character set. By storing these shapes in the buffer, the printer can print them, as if they were part of the normal set.

In bit image (graphics) mode, print resolutions ranging from 480 to 1920 dots per line are possible. This mode allows the printing of shapes not held in the printer's ROM and is used when doing a dump of a HIRES screen. It should be noted that the VZ's screen is only 128 pixels wide. The printer is capable of 15 times this resolution, allowing it to depict much finer detail than is possible on screen.

In the world of printers, the primary standard has been set by EPSON. There can be no doubt of this, since the majority of printer and computer manufacturers produce equipment, which conforms to it. IBM's adoption of the standard is further confirmation. Even the TANDY printers, offer, in their latest models, an IBM-EPSON compatible mode. Your VZ may not be able to compete in the real world, but your printer can. Selecting an EPSON type printer to partner your VZ, means stepping into the mainstream of the computing world. When you finally outgrow your VZ, you will still have a useful peripheral that's able to communicate with almost any computer you happen to choose.

EXTENDED DOS VERSION 1.0 (C) COMMANDS :-

```
MERGE - MERGES basic file from disk with program in memory.
DIRA  -  See example - T:MENU      B:PATCH3.1 B:WORDPROC
                       B:EXTDOS E B:EXTDOS R W:DOS-INST
LDIRA - As above, but to screen and printer.
DIRB  - See example  - T:MENU      01 00 7AE9 801B 0532
                       B:PATCH3.1 01 0B 7200 771F 051F
LDIRB - As above, but to screen and printer.
STATUSA  - Prints free disk space to screen on one line.
LSTATUSA - As above, but to screen and printer, see below.
           534 RECORDS FREE   63.500K FREE
OLD  - Restores a program after using the NEW command.
OLD. - Prints START, END and LENGHT of program in memory in HEX.
DEC XXXXX - Converts DECIMAL to HEX
HEX XXXX  - Converts HEX to DECIMAL
STATUSA and LSTATUSA also works with Version 1.0 DOS.
```

The EXTENDED DOS is available in the two versions below :-
EXTDOS R - T.O.M. SEEKING (SELF RELOCATING)
EXTDOS E - FOR 2K RAM AT 6000-67FF HEX
Price - $10.00 each or the two for $15.00. Availble from :-

Dave MITCHELL - (079) 27 8519
24 ELPHINSTONE STREET NORTH ROCKHAMPTON QUEENSLAND 4701

FOR INFORMATION IN NEWCASTLE AREA :- Joe LEON - (049) 51 2756

FOR SALE - DATABASE - DISK / TAPE

DATA - 16k - VZ DATABASE. Enter data into records thirty characters long (accepts graphic characters). Runs on VZ 200+16k or VZ 300. Available on disk as DISK DATABASE or on tape as CASSETTE DATABASE.

Facilities include data entry into record of choice, into last record chosen, next record, auto-next for fast data entry, edit keys so you don't have to re-enter entire content of a record, delete a record, delete a block of records, gap delete, insert, gap insert, fast alphabetical sort of records--start anywhere in records ; number sort ; swap any two records ; page display--ten records per page ; display current page, next page, previous page, flip backward and forward through datafile, swap any two pages, fast search of entire datafile for a sequence of characters--anywhere in records, hardcopy your records--especially suited for VZ printer plotter ; menu etc.

Disk DATA has Directory and ERASE commands, saves a datafile or any part thereof as a single binary file which loads back quickly. Cassette DATA CSAVES a datafile as a single T file--no slow loading of multitudes of D files! All instructions for using DATA are stored on disk and tape as datafiles--run DATA, load an instruction file and page through it. This program certainly stands out amongst the crowd of other such programs of it's type.

PRICE - $20.00 for DISK or CASSETTE DATABASE - Please make all Cheques and Money Orders payable to and is available from :-
SCOTT LE BRUN 5 CAMERON COURT WANTIRNA VIC. 3152